Capstone Project Report Shabnam Rafat Ummey Sawda

Machine Learning Engineer Nanodegree December 3, 2020

Dog Breed Classifier using Convolutional Neural Network

Domain Background:

Convolutional Neural Network(CNN) is used to analyze visual imagery. Dog Breed Classifier model can be designed using CNN. A model will be built where a number of real world images will be supplied and it will be able to identify a dog's breed. When a dog image will be given as input, it will be able to detect it's breed. In case, a human photo is given as input, it will detect the resembling dog breed. We will use Supervised Learning algorithm here for CNN. At the end, a web app will be built which will predict the breed of a user's input image.

Problem Statement:

The objective is to build a supervised machine learning model that will be able to identify a dog's breed when a user will input a real world image as input. The algorithms will be designed in below ways,

1. If a user input a dog's image, it will process that image and identify the dog's breed.
2. If a user input a human image, it will process that image and identify the resembling dog's breed.

Metrics:

The dataset is split into three section. One is used to train the model; another is to test and the rest one is used to validate dataset. Through the testing dataset, the performance of the built model on unseen data is predicted. The metric using here is "Accuracy". Here,

Accuracy = Number of correctly predicted items/All classified items

Also, during model training, a comparison is made between the test data prediction and validation dataset and Multi class log loss is calculated for finding the best performing model. Log loss best evaluates and quantifies the uncertainty of a measurement based on how much it varies from actual label.

Data Exploration:

In this project the input format is an image format, as the user will input an image and get to know it's breed. To design convolutional neural network, a large dataset is required. Here, the dataset is provided by Udacity. This dataset contains human and dog images.

Dataset of Dog Images: This dataset contains 8351 total images that is sorted into below ways,

Training Directory: *6,680 Images*

Testing Directory : *836 Images*

Valid Directory : *835 Images*

In each directory, there are 133 folders corresponding to dog breeds. The sizes of the images are different. The backgrounds also vary. Some images are cropped ones. The data is not balanced, as the number of images provided for each breed are not same.

Dataset of Human Images dataset: This dataset has 13233 total human images. The sorting is done as by names of human in 5750 folders. The size of each image is 250x250. The backgrounds are not same. Angles are also different. The data is not balanced, as the I total image numbers in each image group are not same.

Picture: Human and Dog Image Datasets

Algorithms and techniques:

To develop the model supervised learning algorithm is used. Convolutional Neural Network(CNN) is used to process, analyze images and classify the breeds.

Convolutional Neural Network is a one of the deep learning methods, which is widely used to detect objects from images through visual imagery.

Here, the solution runs in three steps.

1) For detecting human images, existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers are used.
2) For detecting dog-images, e a pretrained VGG16 model is used.
3) Through the above steps when the image is detected as a human or a dog, the image is paased to an CNN. It will then process, analyze the image and make a prediction about the breed that matches the best out of 133 breeds.

Benchmark:

1) The CNN model that is created from scratch must have accuracy level of at least 10%.
2) The model created using transfer learning must have accuracy level of 60% and above.

Data Preprocessing:

At first, all the images in dataset are resized to a certain size(224X224). Then, normalization is applied to those images (all types of datasets). To reduce overfitting of the training data, image augmentation is done. The training data images are rotated and flipped horizontal in random way. Before passing to the model, all the image data is converted into tensor.

Implementation:

A CNN model is built from scratch to solve the stated problem. This model has 3 convolutional layers. All convolutional layers have kernel size of 3 and stride of 1. The first convolutional layer (conv1) is taking 224X224 input image and the third convolutional layer(conv3) produces an output size of 128. ReLU activation function is used here. To reduce the input size by 2, the pooling layer of (2,2) is used. There are two fully connected layers which finally produces 133-dimensional output. To avoid over overfitting, dropout of 0.25 is added.

Refinement:

The built CNN has accuracy of 13%, That means, it meets the benchmarking. But, the accuracy of the model can be improved by using transfer learning. For creating CNN with transfer learning, Resnet101 architecture is selected, which is pre-trained on ImageNet dataset. This architecture is 101 layers deep. The last convolutional output of Resnet101 is fed as input to our model. It is required to add a fully connected layer for producing 133-dimensional. The model has performed well in comparison to CNN from scratch. With just 5 epochs, it has got 75% accuracy.

Model Evaluation and Validation:

Human Face detector:

OpenCV's implementation of Haar feature based cascade classifiers are used to correctly predict human face resemblances. In first 100 images of human face dataset, 98% were detected as human faces and in first 100 images of dog dataset, 17% of human faces detected.

Dog Face detector:

The dog detector function is created using pre-trained VGG16 model. In first 100 images of dog dataset, 100% of dog faces were detected and 1% of dog faces detected in first 100 images of human dataset.

CNN using transfer learning:

The CNN model is created using transfer learning with ResNet101 architecture was trained for 5 epochs, and the final model produced an accuracy of 75% on test data. The model correctly predicted breeds for 632 images out of 836 total images. Accuracy on test data: 75% (680/836).

Justification:

The built model's performance is better than expected. The model is created using transfer learning, where accuracy level is 75%.

Improvement:

The model's prediction accuracy level can be improved by adding more training and test datasets. Also, by performing more image augmentation, overfitting can be avoided and the accuracy level can be improved.